Fuzzy Control Applied to Economic Stabilization Policies

Vasile Georgescu

Faculty of Economic Sciences

University of Craiova

13, A.I.Cuza Str.,

1100 Craiova

ROMANIA

E-mail: vgeo@central.ucv.ro

Abstract: Applying fuzzy control for stabilizing economic processes is a promising area. On the one hand, fuzzy control could provide a method to implement stabilization strategies in a user-friendly way, by means of a linguistically expressed algorithm. On the other hand, it is more appropriate to deal with nonlinear processes, due to the capability of prescribing nonlinear actions and tuning nonlinear controller parameters. Our main goal is to illustrate how some classical stabilization models can be adapted in order to incorporate fuzzy modes of control. First, we briefly describe certain fundamental concepts of fuzzy control. Afterwards, the focus will be on the design of fuzzy linear controllers that emulate conventional modes of control. A procedure for making the linear fuzzy controller gradually nonlinear and for fine-tuning it completes the design. As an application of fuzzy control to economic processes, a fuzzy extension of Phillips' stabilization model is provided in two variants: for a closed economy as well as for an open economy. Matlab is used to implement the Mamdani-like inference systems and a simulation of the control schemata via Simulink takes place.

Keywords: Economic Stabilization Policies, Emulating Conventional Modes of Control by Fuzzy Controllers, Fine-tuning Nonlinear Fuzzy Controllers

Dr. Vasile Georgescu received the following degrees: MSc. degree in Economic Cybernetics from the Academy of Economic Studies, Bucharest in 1981; MSc. degree in Mathematics from the University of Bucharest in 1990; Ph. D degree in Economics from the Academy of Economic Studies, Bucharest in 1996. Between 1992 and 1997 he was a lecturer and since 1998 he has been a senior lecturer in Economic Cybernetics and Econometrics at the Faculty of Economic Sciences, University of Craiova. He has published 8 books and over 50 papers in national and international journals and conference proceedings. Dr. Vasile Georgescu is a member of the International Association for Fuzzy-Set Management and Economy (SIGEF). His current research interests include fuzzy logic, data mining and knowledge discovery, control theory, economic cybernetics, econometrics.

1. Introduction

Building models for applying stabilization policies to economic processes has been for long attempted. Various classical methods have been used: from econometric models based on rational expectations (which assume that shifts in economic policy produce revised expectations of rational agents), to regulatory mechanisms, normally regarded as engineering tools. Arnold Tustin accomplished a pioneering work in this latter direction. In 1953, he published a book significantly titled "*The Mechanism of Economic Systems: An approach to the problem of economic stabilization from the point of view of control system engineering*". A. W. Phillips continued Tustin's work at the London School of Economics. Originally trained as an electrical engineer, he was aware that an economic system is dynamic, with feedback loops and behaviors that are not susceptible to simple cause and effect. In a well- known article published in 1954 and entitled "*Stabilization Policy in a Closed Economy*", he introduced a stabilization mechanism that was explicitly based on a PID mode of control. It is the historical relevance of this model that gives us the main reason why to adapt it for incorporating a fuzzy control mechanism.

First a theoretical background for our approach is provided. Section 2 resumes certain basic concepts of fuzzy control and makes a summary description of the fuzzy control algorithm. In Section 3 the relationship between linear and fuzzy controllers is addressed. Actually, any fuzzy controller can be regarded as a superset of linear controllers. Hence, the former can emulate the latter under certain assumptions. The design procedure that allows us to emulate a conventional PID controller by means of a linear fuzzy controller is introduced in Section 4.

The emulation of a conventional controller is only a starting point for further exploitation of the full capabilities of a fuzzy controller. One of these consists in the possibility of implementing nonlinear modes of control. We can benefit from this property when designing a fuzzy control strategy to stabilize an economic process, by choosing among different shapes of the membership functions, different implication functions, different T-norms and T-conorms to model the logical connectives used in the antecedent of various rules, and by scaling inputs in order to fully exploit the range of universe of discourse. Classical tuning procedures may be applied during the emulation stage, and some fine-tuning procedures in the final design stage, when making the fuzzy controller gradually nonlinear. The consequences on model performance of such specific choices and various tuning parameters may be evaluated by simulation. Section 5 and Section 6 are discussing these topics.

In order to illustrate how the economic stabilization models can be adapted to use fuzzy modes of control instead of conventional ones, we refer to Phillips' stabilization model, which is a well- known pioneering work in this field. There are two variants of this model we refer to:

- A closed economy stabilization variant, where the model is given in terms of input-output description;
- An open economy stabilization variant, where the model is given in terms of state-space description.

As a consequence, different approaches will be made for either case.

With regard to the former, the stabilization mechanism was originally based on a conventional PID mode of control. Therefore, adopting a more flexible design strategy could reside in replacing it by a fuzzy PID controller. Section 7 describes this approach.

As far as the latter variant is concerned, we have to choose a control strategy in accordance with the state-space description of the model. Hence, a state-feedback controller will be used, a case in which there are two alternative design methods to be applied in determining the feedback gain matrix: the first refers to a pole placement technique and the second consists of a linear quadratic optimization technique. The design procedure is introduced in Section 8.

In all these cases, to transfer gains from conventional to fuzzy controllers is to be done first. Subsequently, one can, according to a nonlinear strategy of control, modify some control characteristics. Finally, fine-tuning of the gains attached to the nonlinear fuzzy controller completes the design strategy. Some advantages of fuzzy control and a road-map for this paper are presented.

2. The Fuzzy Control Algorithm

Basically, a fuzzy controller can be regarded as a nonlinear static function, which maps controller inputs onto controller outputs:

$$y = f(x), \qquad x \in X, \; y \in Y$$

where X and Y are the input and the output space, respectively.

Strictly speaking, it is always possible to use a fuzzy rule-based system as universal approximator of any nonlinear mapping. This implies that:

$$\forall\, x \in X, \; \left|F(x) - f(x)\right| < \varepsilon$$

where $F(x)$ is the function to be approximated and ε can be chosen to be arbitrarily small. The approximation is based on interpolation. First, a discretization is needed: fuzzy coverings on both the input and the output space are considered. Subsequently, a fuzzy rule base is defined, each rule having its antecedent in the input covering and its consequent in the output covering. The consequences of all the active fuzzy rules are then inferred via fuzzy inference. Finally, a crisp output is computed using an interpolative mechanism: all the partial fuzzy consequences are combined into an aggregate one and a numerical value is obtained by defuzzification. The image of the input space through the function $f(x)$ results in a control hypersurface.

We are now going to provide a more detailed description of a fuzzy controller.

Consider the input space X as being an N_X -dimensional referential:

$$X = X_1 \times \cdots \times X_i \times \cdots \times X_{N_X}$$

where X_i is the universe of discourse of the input variable x_i.

For any i in $\{1, \ldots, N_X\}$, let us consider a fuzzy covering of the universe of discourse X_i:

$$\{A_{i,j}\}_{j=\overline{1,m_i}}$$

where $A_{i,j}$ are fuzzy sets. Normally, fuzzy partitions are addressed, that is, fuzzy coverings with the additional property:

$$\sum_{j=1}^{m_i} \mu_{A_{i,j}}(x_i') = 1, \quad \forall\, x_i' \in X_i$$

The rule base includes a set of N_r parallel fuzzy rules. Each fuzzy rule is an *if – then* statement, where the antecedent and the consequent consist of fuzzy propositions. The antecedent (also called premise) contains a combination of propositions through the logical connectives of *and* and *or*. Formally, we have:

r_k : **if** x_1 is $A_{1,k}$ **and** ... **and** x_i is $A_{i,k}$ **and**

... **and** x_{N_X} is $A_{N_X,k}$ **then** y is B_k

In practice, fuzzy control is applied using local inferences. That means each rule is inferred and the results of the inferences of individual rules are then aggregated. According to this approach, the inference in fuzzy control is represented by the following steps:

1. Match the fuzzy propositions x_i *is* $A_{i,k}$, used in the premises of fuzzy rules r_k, with the numerical data x_i' (controller inputs):

   $$\alpha_{i,k} = \mu_{A_{i,k}}(x_i')$$

   where $\alpha_{i,k}$ is a numerical value representing the matching. In the case of fuzzy inputs A_i', the matching is normally represented by:

   $$\alpha_{i,k} = hgt(A_i' \cap A_{i,k})$$

   The two cases can be treated uniformly, if proceeding on a fuzzification of the crisp input x_i', by translating it into a fuzzy singleton:

   $$A_i' = fuzz(x_i'), \qquad \text{with} \qquad \mu_{A_i'(x_i)} = \begin{cases} 1 & if\ x_i = x_i' \\ 0 & otherwise \end{cases}$$

   Thus, $\alpha_{i,k}$ can be obtained using a sup-min composition, which consists of a projection (sup) and a combination (min):

   $$\alpha_{i,k} = \sup_{x_i} \min\left(\mu_{A_i'}(x_i),\ \mu_{A_{i,k}}(x_i)\right)$$

2. Determine the degrees of fulfilment (DOF) β_k for each rule r_k:

   $$\beta_k = \mathop{T}_{i=1}^{N_X} \alpha_{i,k}$$

   where T is the T-norm representing the *and* connective in the premises of the rules. Normally T is chosen as either the *min* or the *product* operator. If the *or* connective is used, this T-norm has to be replaced by a T-conorm. Of course, both the *and* and the *or* connective can be used in the same premise. In such case some values $\alpha_{i,k}$ have to be combined by means of a T-norm (*and* connective), others by means of a T-conorm (*or* connective).

3. Determine the result B'_k of each individual rule r_k:

$$\mu_{B'_k}(y) = I(\beta_k, \mu_{B_k}(y))$$

where I is the implication used to model the fuzzy rule. This can be one of the suitable fuzzy implications. In fuzzy control, a conjunction-based fuzzy implication (also called T-implication) is normally used: either the *min* implication (introduced by Mamdani) or the *product* implication (introduced by Larsen).

4. Aggregate the partial results B'_k of the individual fuzzy rules r_k into the overall result B':

$$\mu_{B'}(y) = \bigcup_k \mu_{B'_k}(y)$$

5. Finally, obtain a crisp output, which a defuzzification method is needed for, starting from the aggregated fuzzy result B'. For example, we can use the center-of-gravity defuzzification method, which is defined by:

$$cog(B') = \frac{\int_Y \mu_{B'}(y) \cdot y \, dy}{\int_Y \mu_{B'}(y) \, dy}$$

Some discrete versions of this method are mostly used in fuzzy control.

The most common inference methods are: the *max-min* method, the *max-product* method and the *sum-product* method, where the aggregation operator is denoted by either *max* or *sum*, and the fuzzy implication operator is denoted by either *min* or *prod*.

The fuzzy controller introduced by Assilian and Mamdani (1974) was based on the *max-min* method. Thus, choosing the *min* operator for conjunction in the premise of rules as well as for the implication function, and the *max* operator for the aggregation, the compositional rule of inference application results in:

$$\mu_{B'}(y) = \max_k \min(\beta_k, \mu_{B_k}(y))$$

with:

$$\beta_k = \min_i \alpha_{i,k};$$

$$\alpha_{i,k} = \sup_{x_i} \min(\mu_{A'_i}(x_i), \mu_{A_{i,k}}(x_i))$$

Notice that the inferred fuzzy consequences B'_k, defined by the membership functions $\mu_{B'_k}(y) = \min(\beta_k, \mu_{B_k}(y))$, are obtained by *clipping* the initial fuzzy sets B_k, due to the *min* operator.

The *max-product* method uses Larsen's implication operator. This inference method is characterized by *scaling* (due to the *product* operator) the consequent B_k of a fuzzy rule r_k with the degree of fulfilment β_k of that rule and by *aggregating* the results B'_k to obtain the fuzzy controller output by means of a *max* operator:

$$\mu_{B'}(y) = \max_k \beta_k \cdot \mu_{B_k}(y)$$

with:

$$\beta_k = \prod_i \alpha_{i,k};$$

$$\alpha_{i,k} = hgt(x'_i \cap A_{i,k}) = hgt(x'_i \cdot A_{i,k}) = \mu_{A_{i,k}}(x'_i)$$

for numerical inputs x_i'. When fuzzy inputs A_i' are faced with, the problem of determining the operator *m* in the *sup-m* composition allows no longer a trivial solution.

The *sum-product* method is similar to the *max-product* method, except for the aggregation operator, which is done by means of summation:

$$\mu_{B'}(y) = \sum_k \beta_k \cdot \mu_{B_k}(y)$$

Also the use of a *bounded sum* is possible, resulting in:

$$\mu_{B'}(y) = \min\left(\sum_k \beta_k \cdot \mu_{B_k}(y),\ 1 \right)$$

which eliminates the occurrence of supernormal fuzzy sets and thus conforms to fuzzy set theory.

3. The Linear Controller As A Subset of A Fuzzy Controller

The emulation of a linear controller can provide an initial fuzzy controller. The latter can further be used as a starting point in designing a more complex fuzzy controller (a nonlinear one), by gradually modifying the initial choices as to the shape of membership functions, logical operators, etc. Only are numerical inputs considered.

In the case of a linear controller the input-output mapping is seen as a linear algebraic equation:

$$y = \sum_{i=1}^{N_x} c_i \cdot x_i + d = c' \cdot x + d$$

where d is an offset. The fuzzy controller function $y = f(x)$ can emulate the linear controller $y = c' \cdot x + d$ when ascertaining the following assumptions:

A1. Membership functions of the fuzzy sets in the universe of discourse of the inputs are triangularly shaped and normal;

A2. The fuzzy sets for each input form a fuzzy partition:

$$\{A_{i,j}\}_{j=\overline{1,m_i}}; \quad \sum_{j=1}^{m_i} \mu_{A_{i,j}}(x_i') = 1, \quad \forall x_i' \in X_i$$

A3. The fuzzy rule base is complete;

A4. A T-norm is used for the implication function (T-implication);

A5. The operator for conjunction in the premises of the fuzzy rules is the *product* operator;

A6. The *(bounded) sum* operator is used for aggregation and for the *or* connective if it is;

A7. Crisp consequents for the individual fuzzy rules are considered and their choice is made in accordance with the linear controller equation $y = c' \cdot x + d$;

A8. The fuzzy-mean defuzzification method is used (which implies the choice of the aggregation operator in A6).

Assumptions A1 to A4 imply that there exists a fuzzy rule for every input combination. Assumptions A5 and A6 (using the *summation* and *product* operators instead of the *max* and *min* operators) are necessary because of the linear controllers emulation requiring operators that result in linear interpolation. The most important assumption to be met is A7, because, given the rest of the assumptions, the output of a fuzzy controller with N_X inputs results in an interpolation of the consequences of at most 2^{N_X} active rules in an N_X-dimensional space. Indeed, if each fuzzy partition contains only normal and convex fuzzy sets, then at any value $x_i' \in X_i'$ there are no more than two consecutive overlapping fuzzy sets $A_{i,\ell}$ and $A_{i,\ell+1}$, such that:

$\mu_{A_{i,\ell}}(x_i') > 0$ and $\mu_{A_{i,\ell+1}}(x_i') > 0$

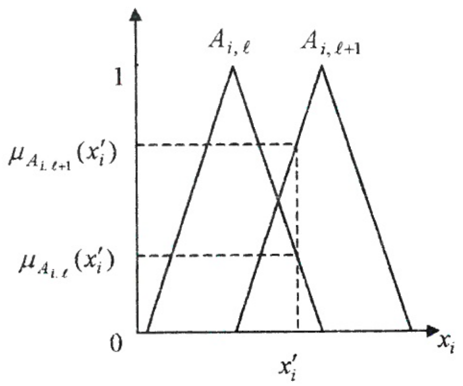


Figure 1. Crisp Input Matching Two Consecutive Overlapping Fuzzy Sets of A Fuzzy Partition

The numerical consequences of at most 2^{N_X} contributing fuzzy rules determine whether or not a linear relation interpolating these points (hyperplane), does exist. When this hyperplane exists, the input-output mapping of the fuzzy system satisfies the linear equation $y = c' \cdot x + d$.

4. Emulating A PID Controller By Means of A Linear Fuzzy Controller

An easy and usually convenient way to start design a nonlinear fuzzy controller is to emulate a conventional PID (proportional-integral-derivative) controller by means of a linear fuzzy controller and to make it progressively nonlinear.

In the case of a conventional PID controller, the control variable $u(t)$ is defined in terms of deviations from or errors $e(t)$ in a reference value y_{ref} and the process output $y(t)$ (i.e. $e(t) = y_{ref} - y(t)$):

$$u(t) = G_p \cdot e(t) + G_i \cdot \int_0^t e(\tau)\, d\tau + G_d \cdot \frac{de(t)}{dt}$$

where G_p, G_i and G_d are proportional, integral and derivative gains, respectively.

If we are concerned with digital control, discrete approximation of the previous relation should be done, and it can be done by replacing the derivative term by a backward difference and the integral by a sum using rectangular integration:

$$u_t = G_p \cdot e_t + G_i \cdot \sum_{\tau=1}^{t} e_\tau T_s + G_d \cdot \frac{e_t - e_{t-1}}{T_s} =$$
$$= G_p \cdot e_t + G_i \cdot ie_t + G_d \cdot ce_t$$

where T_s is the sampling period (in economic applications, T_s is normally assumed to be 1).

In order to emulate the conventional PID controller through a linear fuzzy controller, we have to replace the summation in PID control by a fuzzy rule base acting like a *summation*. The closed loop system should thus show exactly the same step response (this is to check the correctness of implementation).

Hence, a fuzzy PID (FPID) controller uses variables as *error* e, *change of error* ce, and *integral error* ie in the antecedent of *if - then* rules and the *control* variable u (which may be replaced by *change of control* cu, when having to deal with incremental control) in the consequent of rules.

A fuzzy controller based on the Mamdani-type fuzzy inferences would consist of rules having the form:

r_k : **if** (e is $A_{1,k}$) **and** (ce is $A_{2,k}$) **and** (ie is $A_{3,k}$) **then** (u is B_k)

A PID-like fuzzy controller based on Sugeno-type fuzzy inferences has rules of the form:

r_k : **if** (e is $A_{1,k}$) **and** (ce is $A_{2,k}$) **and** (ie is $A_{3,k}$) **then** $u = a_1 \cdot e + a_2 \cdot ce + a_3 \cdot ie$

We turn now to the representation of a fuzzy controller as an input-output mapping. In the general case it may result in a nonlinear shaped control hypersurface. When three inputs (e, ce, ie) and one output (u) are considered, this mapping becomes:

$$u = f(e, ce, ie)$$

However, the assumptions introduced in Section 3 allow us to design a fuzzy rule base acting like a *summation* and resulting in a linear mapping:

$$u_t = Gp \cdot e_t + Gi \cdot ie_t + Gd \cdot ce_t$$

In conventional control, the gains are mainly used for tuning the response. In fuzzy control, scaling of inputs onto a standard universe of discourse is also important and urges to introduce one more parameter to deal with.

So, the next step in the design procedure is to transfer the three gains (Gp, Gi and Gd) used in the conventional PID controller to four gains (say FGp, FGi, FGd and FGu) that are necessary for tuning and scaling the FPID controller. The latter emulates the former if the following condition is met:

$$\begin{aligned} u_t &= Gp \cdot e_t + Gd \cdot ce_t + Gi \cdot ie_t = \\ &= \left[FGp \cdot e_t + FGd \cdot ce_t + FGi \cdot ie_t \right] \cdot FGu = \\ &= FGp \cdot FGu \cdot e_t + FGd \cdot FGu \cdot ce_t + FGi \cdot FGu \cdot ie_t \end{aligned}$$

Comparing the gains of the FPID controller with the gains of the conventional PID controller, the following relations can be derived:

$$FGp \cdot FGu = Gp \quad \Rightarrow \quad FGu = \frac{1}{FGp} \cdot Gp$$

$$FGd \cdot FGu = Gd \quad \Rightarrow \quad FGd = FGp \cdot \frac{Gd}{Gp}$$

$$FGi \cdot FGu = Gi \quad \Rightarrow \quad FGi = FGp \cdot \frac{Gi}{Gp}$$

Let us now assume that the error is in the range $[-E, E]$ and for the fuzzy controller we set a standard input universe, say [-100, 100]. In such a case, we have:

$$e \in [-E, E] \quad \Rightarrow \quad FGp \cdot e \in [-FGp \cdot E, FGp \cdot E] = [-100, 100]$$

Thus, we can set:

$$FGp = \frac{100}{E}$$

The other gains are now fixed as follows:

$$FGu = \frac{E}{100} \cdot Gp \, ; \qquad FGd = \frac{100}{E} \cdot \frac{Gd}{Gp} \, ; \qquad FGi = \frac{100}{E} \cdot \frac{Gi}{Gp}$$

The conventional PID controller may be tuned using the Ziegler-Nichols frequency response method, resulting in optimal values for the parameters Gp, Gi and Gd. Afterwards, the equivalent FPID controller is obtained, deriving its parameters (FGp, FGi, FGd and FGu) from those of the PID controller.

The FPID controller scheme is shown in Figure 2.

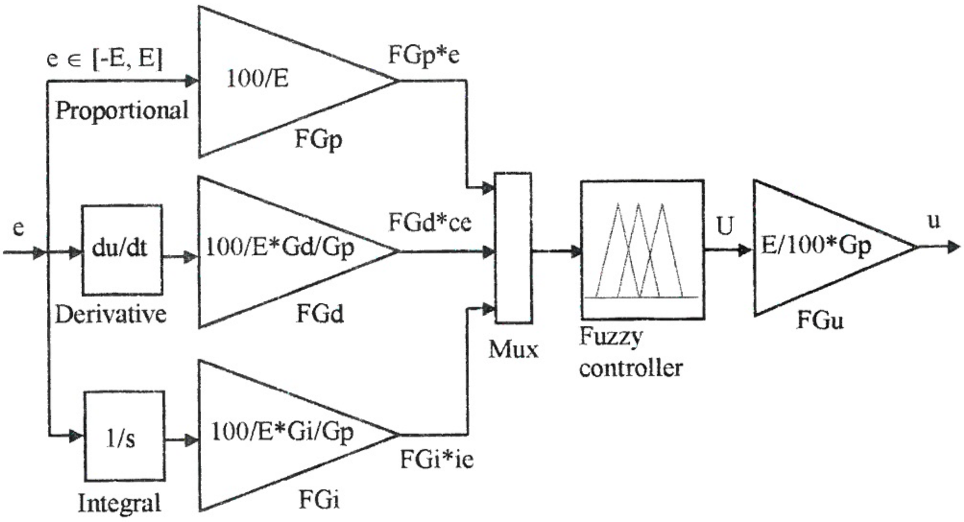


Figure 2. Fuzzy PID Controller (FPID)

Sometimes, a rule base with only two inputs is more convenient. An easy way to reducing the number of fuzzy controller inputs is to separate the integral action as in the fuzzy PD+I (FPD+I) controller shown in Figure 3.

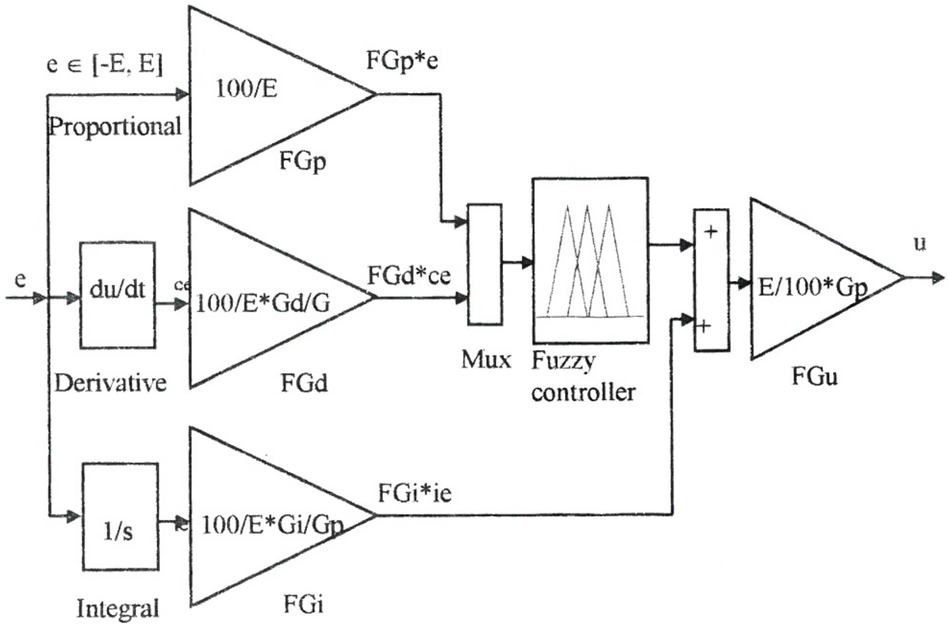


Figure 3. Fuzzy PD+I (FPD+I) Controller

The controller function is thus split into two additive parts:

$$u = u_{FPD} + u_I = f_{FPD}(e, ce) + f_I(ie)$$

The first one corresponds to the FPD controller and is represented by a surface. At the initial stage of the design, this surface is a plane, but it may be rendered nonlinear at a subsequent stage. For example, we can modify the shape of the surface by manipulating the membership functions. Its associated plot is a design aid by visual inspection when selecting membership functions and constructing rules.

Now let illustrate the construction of a fuzzy rule base for a FPD+I controller.

We consider a standard universe of discourse (say $[-100, 100]$) for both inputs: error and change of error. For the sake of simplicity, the same fuzzy partition will be considered in both cases (see Figure 4, where "N", "AZ" and "P" stand for "Negative", "About Zero" and "Positive", respectively).

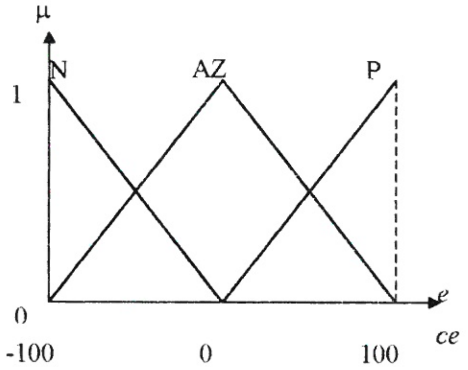


Figure 4. Fuzzy Partition on the Standard Universe of Discourse [-100, 100]

Due to the summation, the standard universe for the output variable u will be $[-200, 200]$. According to assumption A7 introduced in Section 3, fuzzy singletons (whose positions are determined by the sum of the peak positions of the input sets) will be chosen as consequents to fuzzy rules (Figure 5).

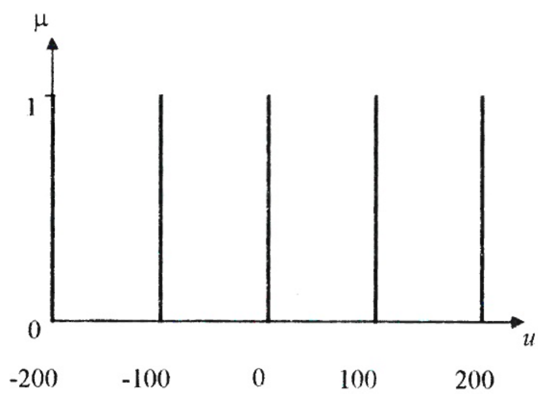


Figure 5. Fuzzy Singletons Chosen As Consequents to Fuzzy Rules

On choosing the design parameters according to assumptions A1 through A8, the control surface degenerates to a diagonal plane (see Figure 6).

The following Matlab program implements these choices, resulting in a fuzzy inference system (FIS):

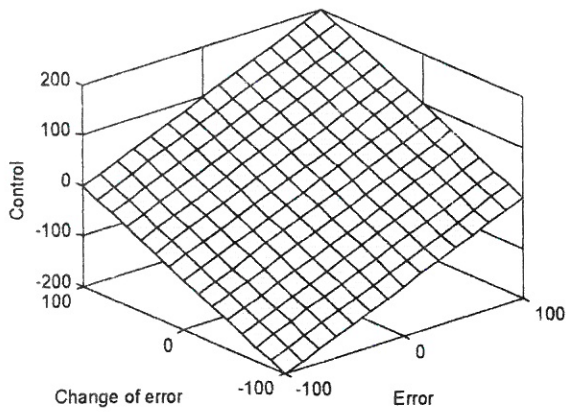


Figure 6. Linear Control Surface

```
% Preliminary computation
standard_input_range = [-100 100];
number_of_inputs = 2;
number_of_input_mfs = 3;
min_ir = min(standard_input_range);
max_ir = max(standard_input_range);
input_tri_mfs = [-100 -100  0;  -100  0  100;   0  100  100];
step = (max_ir - min_ir) / (number_of_input_mfs - 1);
min_out_range = number_of_inputs * min_ir;
max_out_range = number_of_inputs * max_ir;
standard_output_range = [min_out_range, max_out_range];
output_tri_mfs = [];
for v = min_out_range: step: max_out_range
  current_mf = [v v v];      % degenerated triangle = singleton
  output_tri_mfs = [output_tri_mfs; current_mf];
end
number_of_output_mfs = size(output_tri_mfs, 1);

 % Build FIS for FPD controllers
q=newfis('FPD');
q.andMethod = 'prod';
q.impMethod = 'prod';
q.orMethod = 'sum';
q.aggMethod = 'sum';

% Add the first input variable
q = addvar(q, 'input', 'Error', standard_input_range);
q = addmf(q, 'input', 1, ' Negative ', 'trimf', input_tri_mfs(1,:));
q = addmf(q, 'input', 1, 'About Zero', 'trimf', input_tri_mfs(2,:));
q = addmf(q, 'input', 1, ' Positive ', 'trimf', input_tri_mfs(3,:));

% Add the second input variable
q = addvar(q, 'input', 'Change of error', standard_input_range);
q = addmf(q, 'input', 2, 'Negative', 'trimf', input_tri_mfs(1,:));
q = addmf(q, 'input', 2, 'About Zero', 'trimf', input_tri_mfs(2,:));
q = addmf(q, 'input', 2, 'Positive', 'trimf', input_tri_mfs(3,:));

% Add the output variable
q = addvar(q, 'output', 'Control', standard_output_range);
q = addmf(q, 'output', 1, 'Neg_Big', 'trimf', output_tri_mfs(1,:));
q = addmf(q, 'output', 1, 'Neg_Small', 'trimf', output_tri_mfs(2,:));
```

```
q = addmf(q, 'output', 1, 'Zero', 'trimf', output_tri_mfs(3,:));
q = addmf(q, 'output', 1, 'Pos_Small', 'trimf', output_tri_mfs(4,:));
q = addmf(q, 'output', 1, 'Pos_Big', 'trimf', output_tri_mfs(5,:));

% Add the rules. Fuzzy singletons (whose positions are determined by the sum
% of the peak positions of the input sets) are chosen as consequents
ruleList = [];
for i = 1: number_of_input_mfs
  for j = 1: number_of_input_mfs
    output_val = input_tri_mfs(i, 2) + input_tri_mfs(j, 2);
    k = (output_val - min_out_range)/step + 1;
    current_rule = [i  j  k  1  1];
    ruleList = [ruleList; current_rule];
  end
end

q = addrule(q,ruleList);
surfview(q)
```

5. Making the Linear Fuzzy Controller Progressively Nonlinear

There are three sources of nonlinearity in a fuzzy controller:

- *The rule base.* The position, the shape and the number of fuzzy sets as well as the nonlinear input scaling cause nonlinear transformations. The rules often express a nonlinear control strategy.
- *The inference engine.* If the connectives *and* and *or* are implemented as for example *min* and *max*, respectively, they are nonlinear.
- *The defuzzification.* Several defuzzification methods are nonlinear.

All the characteristics enumerated above can be used to gradually make the linear fuzzy controller nonlinear. The shape of the sets and the choice of rules can most easily be applied.

The shapes of the control surface induced by certain input families of fuzzy sets are shown in Figure 7.

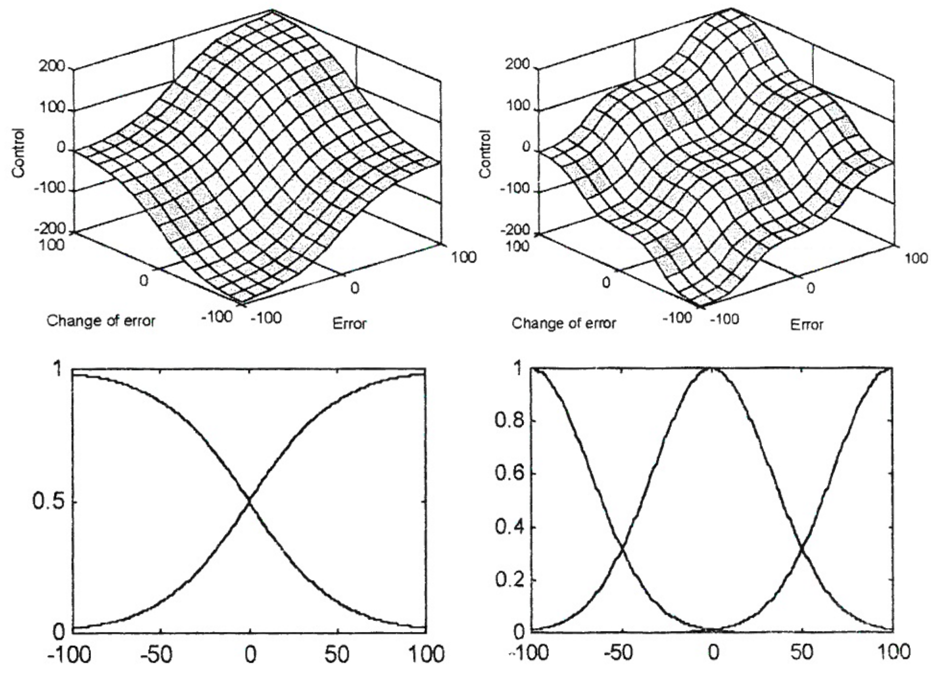


Figure 7. Nonlinear Control Surfaces Induced By Different Input Families

The control characteristics producing nonlinearities that affect the closed-loop system dynamics must be chosen by analysing how the response reacts to each of them.

6. Fine-tuning of the Nonlinear Fuzzy Controller

The final phase in the design procedure is that of fine-tuning the gains attached to the nonlinear fuzzy controller. This operation combines the intuition with some rules of thumb derived from experience. In simulation it is possible to experiment with different control surfaces and to get an idea of the gain margin and of the characteristics of response.

There is a large potential for the design approach in fuzzy PID controllers, due to a widespread application of PID control and well-known tuning rules.

A procedure for hand-tuning of an FPD+I controller may be sketched as follows:

1. Adjust FGp according to the reference step size and the universe of discourse in order to fully exploit the range of the input universe.

2. Remove integral action and derivative action by setting $FGd = FGi = 0$. Tune FGu to give the desired response, ignoring any final value offset.

3. Increase the proportional gain by means of FGu, and adjust the derivative gain by means of FGd to dampen the overshoot.

4. Adjust the integral gain by means of FGi to remove any final value offset (steady state errror).

5. Repeat the whole procedure until FGu gets as large as possible.

7. Phillips' Stabilization Model for A Closed Economy and Its Fuzzy Extension

7.1 Conventional (PID-like) Modes of Control

Phillips' model naturally assumes that the level of aggregate demand determines the level of national income. The former is made up of a part originating from private economic agents and a part originating from the government. The stabilization policy consists in the adjustment of government expenditure in order to increase or decrease the aggregate demand, resulting in a desired level of the national income. Faced with modifications induced in aggregate demand, the producers react by making some adjustments in output: if aggregate demand exceeds the current output, the latter will be increased; otherwise, it will be decreased.

The stabilization model is given by the equations:

$$\dot{Y}(t) = a \cdot (D(t) - Y(t)); \qquad a > 0 \tag{1}$$
$$D(t) = (1-\ell) \cdot Y(t) + G(t) - v; \quad 0 < \ell < 1 \tag{2}$$
$$\dot{G}(t) = b \cdot (G^*(t) - G(t)); \qquad b > 0 \tag{3}$$
$$G^*(t) \text{ - given in the form of a control policy} \tag{4}$$

where: $Y(t)$ is the national income;

$D(t)$ is the aggregate demand;

a is a reaction coefficient (representing the velocity of adjustment to a discrepancy between aggregate demand and current output);

$(1-\ell)$ is the marginal propensity to spend (i.e. the marginal propensity to consume plus the marginal propensity to invest);

v is an exogenous disturbance, indicating a decrease in aggregate demand;

$G(t)$ is the actual government demand;

$G^*(t)$ is the potential government demand, which stands for the stabilization policy;

b is a reaction coefficient, indicating the speed of response to a discrepancy between potential and actual public expenditure.

In order to simplify the analysis, the variables are measured in terms of deviation from their desired levels, so that a negative value simply means that the actual value is smaller than the desired value. Thus, we are led to consider the reference value Y_{ref} as being 0 and the error as being:

$$e(t) = Y_{\text{ref}} - Y(t) = -Y(t) \tag{5}$$

Let us assume that national income is initially at the desired level and that an exogenous decrease in aggregate demand occurs.

The stabilization policy:

$$u(t) = G^*(t) \tag{6}$$

proposed by Phillips, was defined in terms of a proportional (P), or derivative (D), or integral (I) mode of control, or a combination of these three modes, that is in terms of a PID control policy. In the latter case, the relation defining the control variable takes the form:

$$u(t) = G_p \cdot e(t) + G_i \cdot \int_0^t e(\tau)\, d\tau + G_d \cdot \frac{de(t)}{dt} \tag{7}$$

where G_p, G_i and G_d are the proportional, integral and derivative gains, respectively.

Phillips' model can be manipulated in order to reduce it to a single equation. Finally we obtain:

$$\ddot{Y} + (a\ell + b)\dot{Y} + ab\ell Y - abG^* = -abv \tag{8}$$

When we consider one unit decrease $v = 1$ in aggregate demand, the differential equation of the model becomes:

$$\ddot{Y} + (a\ell + b)\dot{Y} + ab\ell Y - abG^* = -ab \tag{9}$$

Inserting the various modes of control defining $u(t) = G^*(t)$ as generically expressed by (7), let us determine the time path of national income.

The conventional PID control scheme for Phillips' stabilization model is shown in Figure 8:

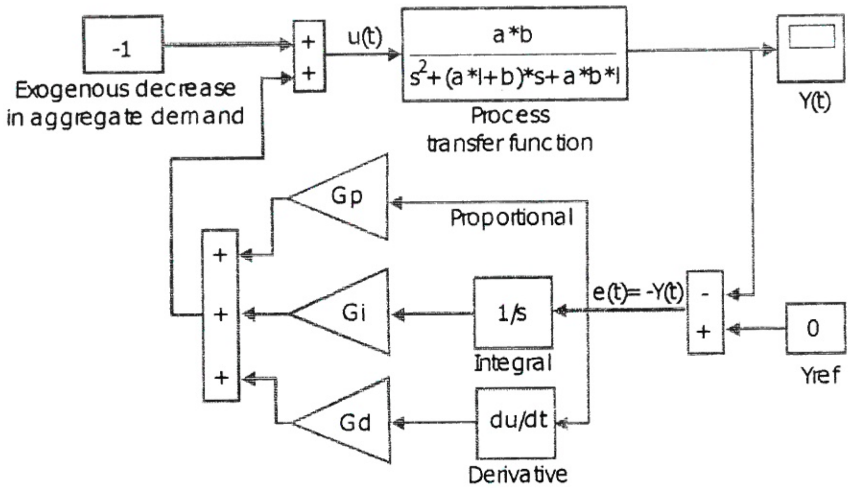


Figure 8. Control Scheme for the Classical Phillips Model

It can easily be shown that, without any stabilization policy, the limit value of national income $Y(t)$ (measured in terms of deviation from its desired level) is $y^* = -1/\ell$ (see curve (1) in Figure 9a).

When only a proportional stabilization policy is used (i.e. $G_p > 0, G_i = G_d = 0$), the steady state solution becomes:

$$y^* = -1/(\ell + G_p) \qquad (10)$$

and it is smaller in absolute value. That means that the decrease in income determined by an exogenous decrease in aggregate demand is smaller than the decrease occurring in default of a stabilization policy. However, a purely proportional stabilization policy would fail to completely stop the reduction in income and would tend to provoke oscillations, though damped, when G_p is too great. Moreover, the characteristic equation has:

- real roots for $G_p \le \frac{(a\ell - b)^2}{4ab}$, in which case the time path of national income is monotonic;
- complex roots for $G_p > \frac{(a\ell - b)^2}{4ab}$, in which case the time path of national income is oscillatory.

The addition of a derivative policy G_d to the proportional one has no effect as far as the reduction in income is concerned (i.e. the steady state solution remains $y^* = -1/(\ell + G_p)$), but offsets the bias towards oscillations of G_p. One can see that the real part of the characteristic roots is $-\frac{1}{2}(a\ell + b + ab\,G_d)$ in the case of the mixed proportional-derivative policy, and $-\frac{1}{2}(a\ell + b)$ in the case of the purely proportional one. Thus, the former is obviously greater in absolute value than the latter.

A pure integral policy proves successful when G_i is smaller than a certain critical value (i.e. $G_i < (a\ell + b)\ell$), but may provoke instability in the opposite case. If successful, it has the advantage of being able to completely done away with the effects of exogenous disturbance.

When the integral and the proportional stabilization policies are used together, the critical stability limit is greater than in the case of a pure integral policy (i.e. $G_i < (a\ell + b)(\ell + G_p)$), the conclusion being that the addition of the proportional policy contributes to running out of the danger of explosive oscillations.

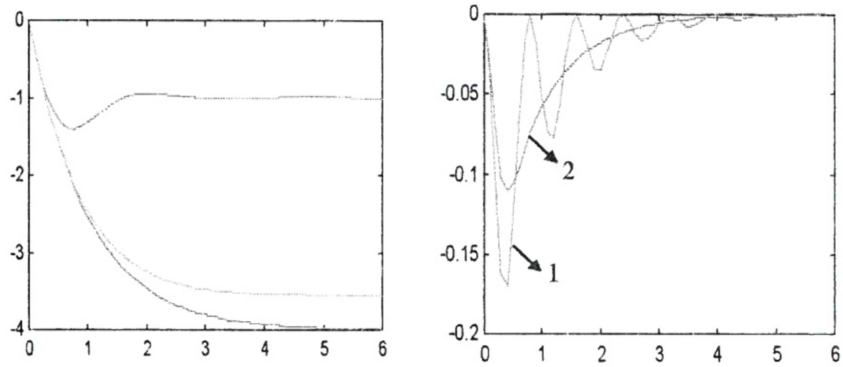
In the case of a mixed integral-derivative policy, the crucial stability condition is $G_i < (a\ell + b + abG_d)\ell$ and provided all the three policies are simultaneously adopted, the crucial stability condition is

$$G_i < (a\ell + b + abG_d)(\ell + G_p).$$

Let us assume the following parameters for the model described above: $a = 4$, $b = 2$, $\ell = 0.25$.

In Figure 9a, curve (1) depicts the time path of national income (measured in terms of deviation from its desired level), without any stabilization policy (there is a stationary error of -4). Curve (2) depicts the case where $G_p = \frac{(a\ell - b)^2}{4ab}$, indicating the frontier between a monotonic behaviour and an oscillatory one. Curve (3) illustrates the case when a steady state solution of $y^* = -1$ is desired, involving $G_p = 1 - \ell = 0.75$.

The time paths in Figure 9b show that the effect of an integral policy is to obtain a null stationary error, while the effect of a derivative policy is to offset the oscillatory bias of other policies.



a. (1) $G_p = G_i = G_d = 0$;
(2) $G_p = 1/32$; $G_i = G_d = 0$;
(3) $G_p = 0.75$; $G_i = G_d = 0$

b. (1) $G_p = G_i = 8$; $G_d = 0$;
(2) $G_p = G_i = 8$; $G_d = 1$

Figure 9. Time Paths for Specific Choices of Controller Gains

7. 2 The Linear Fuzzy Controller That Emulates the PID-like Phillips' Stabilization Model

The design procedure of the FPID controller was exposed in detail in the previous Sections. So, we are now in a position to represent the scheme of this fuzzy controller. The scheme is shown in Figure 10, where the input range $[-E, E]$ was set to $[-4, 4]$.

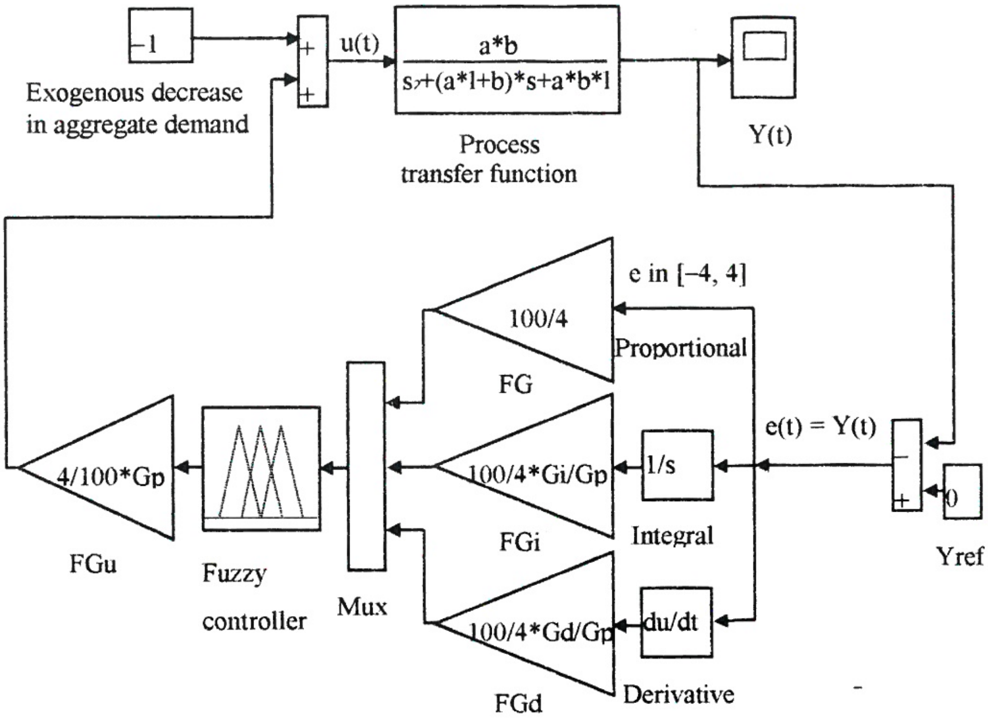


Figure 10. Fuzzy PID Control Scheme for Phillips' Stabilization Model

8. Phillips' Stabilization Model for An Open Economy and Its Fuzzy Extension

8.1 Control Scheme Using A Conventional State-feedback Controller

Let us now modify Phillips' model so that foreign trade flows in economy are taken into account.

We shall consider once again the government expenditure as a control variable. This time the aggregate demand is expressed by the equation:

$$D(t) = G(t) + Z(t) - M(t) + C(t) + I(t) \tag{11}$$

Since the aggregate demand usually differs from the aggregate supply, the stabilization policy has to damp such a difference (expressed by an excess demand $E(t) = D(t) - Y(t)$) whenever it occurs. The following equations complete the model:

$$v \cdot \dot{Y}(t) = I(t) \tag{12}$$
$$I(t) = \sigma \cdot (D(t) - Y(t)) \tag{13}$$
$$C(t) = (1 - s) \cdot Y(t) \tag{14}$$
$$Z(t) = z \cdot Y(t) \tag{15}$$
$$M(t) = m \cdot Y(t) + D(t) - Y(t) \tag{16}$$

where $M(t)$ are imports, $Z(t)$ are exports and $v > 0$, $\sigma > 0$, $0 < s < 1$, $z > 0$, $m > 0$ are constants.

The basic dynamic mechanism of this model is of multiplier-accelerator type (Eq (12) being of accelerator type and Eq (13) of multiplier type).

After successive substitutions of variables $D(t)$, $Z(t)$, $M(t)$, $C(t)$, $I(t)$ in Eqs (11)-(16) and some simple manipulations, the dynamic equation of aggregate supply gets the form of:

$$\dot{Y}(t) = \alpha \cdot Y(t) + \beta \cdot G(t) \tag{17}$$

where:

$$\alpha = \frac{\sigma(z-m-s)}{v(2-\sigma)} \tag{18}$$

$$\beta = \frac{\sigma}{v(2-\sigma)}; \quad \sigma \neq 2 \tag{19}$$

Let us assume that $Y(0) = Y_0$ at $t = 0$. Eq (17) has a unique solution $Y(t)$, for any fixed control strategy $G(t)$.

An *equilibrium solution* is a solution where all quantities $D(t)$, $Y(t)$, $Z(t)$, $M(t)$, $C(t)$, $I(t)$, $G(t)$ are constant as time functions, and the aggregate demand equals the aggregate supply. Thus a necessary condition for the constants $D^* = Y^*, Z^*, M^*, C^*, I^*, G^*$ to designate an equilibrium solution of the above model is to verify the equations:

$$Y^* = G^* + Z^* - M^* + C^* + I^*; \qquad I^* = 0$$
$$C^* = (1-s)\cdot Y^*; \qquad Z^* = z\cdot Y^*; \qquad M^* = m\cdot Y^*$$

Solving this system with respect to G^*, results in:

$$Y^* = \frac{G^*}{s-z-m}; \quad C^* = \frac{(1-s)}{s-z-m}G^*; \quad Z^* = \frac{z}{s-z-m}G^*;$$

$$M^* = \frac{m}{s-z-m}G^*; \quad I^* = 0$$

Let denote by:

$$\underline{D}(t) = D(t) - Y^*; \quad \underline{Y}(t) = Y(t) - Y^*; \quad \underline{G}(t) = G(t) - G^*; \quad \underline{Z}(t) = Z(t) - Z^*$$

$$\underline{M}(t) = M(t) - M^*; \quad \underline{C}(t) = C(t) - C^*; \quad \underline{I}(t) = I(t)$$

the deviations of variables $D(t)$, $Y(t)$, $Z(t)$, $M(t)$, $C(t)$, $I(t)$, $G(t)$ from their equilibrium state. Surely, such deviations satisfy the equations:

$$\underline{D}(t) = \underline{G}(t) + \underline{Z}(t) - \underline{M}(t) + \underline{C}(t) + \underline{I}(t)$$
$$\underline{C}(t) = (1-s)\cdot\underline{Y}(t); \quad v\cdot\dot{\underline{Y}}(t) = \underline{I}(t);$$
$$\underline{Z}(t) = z\cdot\underline{Y}(t); \quad \underline{I}(t) = \sigma\cdot(\underline{D}(t) - \underline{Y}(t));$$
$$\underline{M}(t) = m\cdot\underline{Y}(t) + \underline{D}(t) - \underline{Y}(t)$$

As above, by successive substitutions, the model may be reduced to a single differential equation:

$$\dot{\underline{Y}}(t) = \alpha\cdot\underline{Y}(t) + \beta\cdot\underline{G}(t) \tag{20}$$

where α and β are given by (18) and (19).

$\underline{S}(t)$ denotes the foreign trade balance, cumulated over the time-interval [0, t]:

$$\underline{S}(t) = \int_0^t \underline{S}(\tau)\,d\tau + \underline{S}(0) = \int_0^t \left[\underline{M}(\tau) - \underline{Z}(\tau)\right]d\tau + \underline{S}(0)$$

If differentiating, we have $\dot{\underline{S}}(t) = \underline{M}(t) - \underline{Z}(t)$. To express $\underline{M}(t)$ and $\underline{Z}(t)$ with respect to $\underline{Y}(t)$ and $\underline{G}(t)$ is to obtain:

$$\dot{\underline{S}}(t) = \gamma\cdot\underline{Y}(t) + \delta\cdot\underline{G}(t) \tag{21}$$

where:

$$\gamma = \frac{(m-z)(1-\sigma)-s}{z-\sigma}; \quad \delta = \frac{1}{2-\sigma} \tag{22}$$

The state-space description of the system is finally given by Eq (20) and Eq(21):

$$\begin{cases} \underline{\dot{Y}}(t) = \alpha \cdot \underline{Y}(t) + \beta \cdot \underline{G}(t); & Y(0) = \underline{Y}_0 \\ \underline{\dot{S}}(t) = \gamma \cdot \underline{Y}(t) + \delta \cdot \underline{G}(t); & \underline{S}(0) = \underline{S}_0 \end{cases} \tag{23}$$

With the following notations:

$$x(t) = \begin{pmatrix} \underline{Y}(t) \\ \underline{S}(t) \end{pmatrix}; \quad u(t) = \underline{G}(t); \quad x_0 = \begin{pmatrix} \underline{Y}_0 \\ \underline{S}_0 \end{pmatrix}$$

$$A = \begin{pmatrix} \alpha & 0 \\ \gamma & 0 \end{pmatrix}; \quad B = \begin{pmatrix} \beta \\ \delta \end{pmatrix};$$

the state-space description of the system becomes:

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t); \qquad x(0) = x_0 \tag{24}$$

In order to stabilize the system, we need design a state-feedback controller, to be described by the control law:

$$u(t) = -H\,x(t) \tag{25}$$

The feedback gain matrix $H = (h_1 \quad h_2)$ must be determined such that the closed-loop system $\dot{x}(t) = (A - BH) \cdot x(t)$ is asymptotically stable (i.e. all eigenvalues of $A - BH$ have negative real parts).

There are two alternative design methods: the first is based on a pole placement technique and the second on a linear quadratic optimization technique. In both cases, the design solution can easily be implemented in Matlab.

The state-feedback control scheme is depicted in Figure 11 and can be simulated via Simulink.

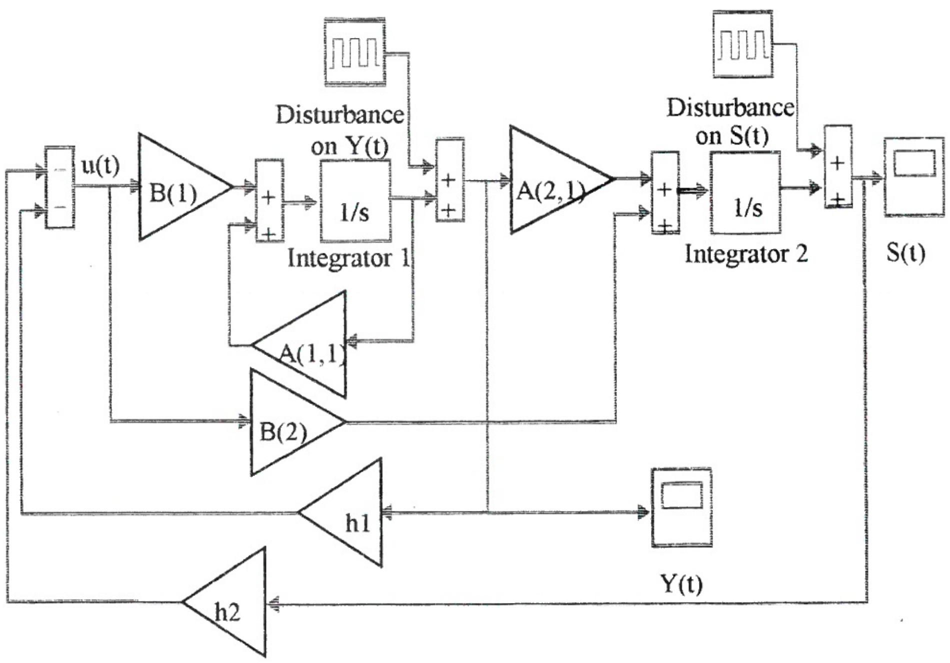


Figure 11. Conventional State-feedback Controller

8.2 The Linear Fuzzy Controller Emulating the Conventional State-feedback Controller

An equivalent fuzzy controller that emulates the conventional state-feedback controller should reproduce the same dynamic behavior of the closed-loop system; this is to check that the implementation is correct.

In our case, the gain matrix H is of the form $H = (h_1 \quad h_2)$ and hence, the control law can be written out:

$$u(t) = H \cdot x(t) = (h_1 \quad h_2) \cdot \begin{pmatrix} \underline{Y}(t) \\ \underline{S}(t) \end{pmatrix} = h_1 \cdot \underline{Y}(t) + h_2 \cdot \underline{S}(t) \tag{26}$$

The feedback gains from the previous control law have to be transferred to the fuzzy controller. However, since the fuzzy controller is normally equipped with input gains as well as with an output gain, this transfer will introduce some degree- of -freedom. More precisely, the fuzzy controller can be written as:

$$u(t) = f(G_1 \cdot \underline{Y}(t),\ G_2 \cdot \underline{S}(t)) \cdot G_3$$

and assuming that it acts as a summation, we obtain the following linear control law:

$$u(t) = (G_1 \cdot \underline{Y}(t) + G_2 \cdot \underline{S}(t)) \cdot G_3 = G_1 \cdot G_3 \cdot \underline{Y}(t) + G_2 \cdot G_3 \cdot \underline{S}(t) \tag{27}$$

Comparing the gains of the conventional state-feedback controller in (26) with the gains of the linear fuzzy controller in (27), the following relations can be determined:

$$h_1 = G_1 \cdot G_3 \quad \Rightarrow \quad G_3 = \frac{1}{G_1} \cdot h_1 \tag{28}$$

$$h_2 = G_2 \cdot G_3 \quad \Rightarrow \quad G_2 = \frac{1}{G_3} \cdot h_2 = G_1 \frac{h_2}{h_1} \tag{29}$$

With three gains to be determined and two equations, there is one degree- of -freedom. If the input universes are chosen to be standard universes $[-100, 100]$, then we have, in order to avoid the saturation in the universes, the following new constraint:

$$|G_1 \underline{Y}(t)|_{max} \le 100$$

Thus from $G_1 \cdot \underline{Y}(t) \in \left[-G_1 \cdot \underline{Y}^*, G_1 \cdot \underline{Y}^*\right] = [100, 100]$, we deduce:

$$G_1 = \frac{100}{Y^*}$$

The other two gains are now fixed as follows:

$$G_3 = \frac{Y^*}{100} \cdot h_1; \quad G_2 = \frac{100}{Y^*} \cdot \frac{h_2}{h_1}$$

With this procedure of transferring gains, the control scheme of fuzzy controller is obtained as in Figure 12.

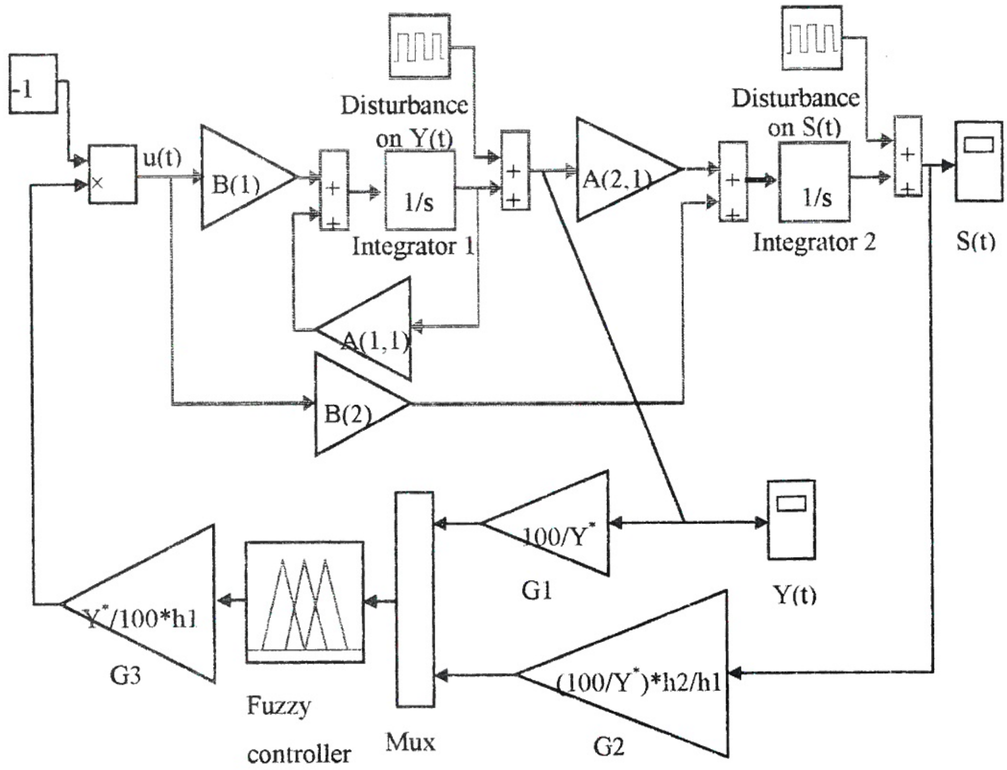


Figure 12. Fuzzy Controller Emulating A Conventional State-feedback Controller

9. Concluding Remarks

When dealing with nonlinear and time-varying behaviour, the stabilization strategies of economic systems are challenging actions. Advanced control techniques have to be addressed in such cases. As fuzzy control can be described as a nonlinear mapping, the corresponding fuzzy controller acts as a nonlinear controller and hence it provides increasing flexibility. Given a fuzzy controller contains a linear controller as a special case, it is true to say that it performs at least as good as the latter. Essentially, a fuzzy controller is a rule-based controller. As a result, the shapes of the control surface can be individually manipulated for different regions of the state space. When the parameters of a conventional controller (a PID one, for example) are tuned, they affect the shape of the entire control surface. In fuzzy control, such possible effects are limited to neighbouring regions only.

This paper primarily focused on the emulation of a conventional controller (either a PID or a state-feedback controller) through a linear fuzzy controller as a starting point for further exploitation of the full capabilities of the nonlinear fuzzy controller. For illustration purposes, we adapted a classical stabilization model (Phillips' model) in order to incorporate fuzzy modes of control. We also suggested how to make the fuzzy controller gradually nonlinear and how to use fine-tuning procedures for achieving the validation objective of the controller. However, handling fuzzy control is rather a complex task, taking much experience. Actually, the potential for performing better depends on the designer capability to exploit the nonlinear options in the fuzzy controller to his best advantage.

BIBLIOGRAPHY

BATUR, C. and KASPARIAN, V., **Predictive Fuzzy Expert Controllers**, COMPUTERS IN INDUSTRIAL ENGINEERING, 20(2), 1991.

DRIANKOV, D., HELLENDOORN, H. and REINFRANK, M., **An Introduction to Fuzzy Control**, SPRINGER-VERLAG, New York, 1993.

DUBOIS, D. and PRADE, H., **Fuzzy Sets and Systems: Theory and Applications**, Vol. 144 , Mathematics in Science and Engineering, ACADEMIC PRESS, 1980.

DUBOIS, D. and PRADE, H., **Possibility Theory - An Approach to Computerized Processing of Uncertainty,** PLENUM PRESS, New York, 1988.

DUBOIS, D. and PRADE, H., **Fuzzy Sets in Approximate Reasoning, Part 1: Inference With Possibility Distributions**, FUZZY SETS AND SYSTEMS, 40, 1991, pp. 143–202.

GUPTA, M. M. and QI, J., **Design of Fuzzy Logic Controllers Based On Generalized t-operators,** FUZZY SETS AND SYSTEMS, 40, 1991, pp. 473-489.

GUPTA, M. M. and QI, J., **Theory of T-norms and Fuzzy Inference Methods**, FUZZY SETS AND SYSTEMS, 40, 1991b, pp. 431–450.

JANTZEN, J., **A Robustness Study of Fuzzy Control Rules**, in EUFIT (Eds.) Proceedings of the 5th European Congress on Fuzzy and Intelligent Technologies, ELITE FOUNDATION, Aachen, 1997, pp. 1222-1227.

HARRIS, C. J., MOORE, C. G. and BROWN, M., **Intelligent Control - Aspects of Fuzzy Logic and Neural Nets**, WORLD SCIENTIFIC, Singapore, 1993.

HOBBS, A. and BOURBAKIS, N. G., **A Neuro Fuzzy Arbitrage Simulator for Stock Investing**, Proceedings of the IEEE / IAFE 1995 Computational Intelligence for Financial Engineering, New York, 1995.

LARSEN, P. M., **Industrial Applications of Fuzzy Logic Control**, INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES, 12(1), 1980, pp. 3–10.

LEE, C.C., **Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I & Part II**, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, 20(2), 1990, pp. 404-435.

LI, M. X., BRUIJN, P. M. and VERBRUGGEN, H.B., **Tuning Cascade PID Controllers Using Fuzzy Logic**, MATHEMATICS AND COMPUTERS IN SIMULATION, 37, 1994, pp. 143-151.

MALKI, H. A., LI, H. and CHEN, G., **New Design and Stability Analysis of Fuzzy Proportional-derivative Control Systems**, IEEE TRANSACTIONS ON FUZZY SYSTEMS 2(4), 1994, pp. 245-254.

MAMDANI, E. H. and ASSILIAN, S., **An Experiment in Linguistic Synthesis With A Fuzzy Logic Controller**, INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES, 7, 1975, pp. 1–13.

MIZUMOTO, M., **Realization of PID Controls by Fuzzy Control Methods**, in IEEE (Eds.) 1st International Conference on Fuzzy Systems, The Institute of Electrical and Electronics Engineers, Inc., San Diego, 1992, pp. 709-715.

PEDRYCZ, W. , **Fuzzy Control and Fuzzy Systems,** 2nd, extended, edition, Taunton, England, RESEARCH STUDIES PRESS LTD., 1993.

PHILLIPS, A.W. , **Stabilization Policy in A Closed Economy**, ECONOMIC JOURNAL, 64, 1954, pp. 290-323

SUGENO, M. and NISHIDA , M., **Fuzzy Control of Model Car**, FUZZY SETS AND SYSTEMS, 16, 1985, pp. 103–113.

TSOUKALAS, L. and UHRIG, R., **Fuzzy and Neural Approaches in Engineering**, JOHN WILEY & SONS, New York, 1997.

TZAFESTAS, S. G. and PAPANIKOLOPOULOS, N. P., **Incremental Fuzzy Expert PID Control**, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 37(5), 1990, pp. 365–371.

S. G. Tzafestas, and A.N. Venetsanopoulos (Eds.) **Fuzzy Reasoning in Information, Decision and Control Systems**, KLUWER ACADEMIC PUBLISHERS, 1994.

YAGER, R.R. , **Alternative Structures for Knowledge Representations in Fuzzy Logic Controllers**, in A. Kandel and Langholz (Eds.) Fuzzy Control Systems, Chapter 5, CRC PRESS, 1994, pp. 99-137.

ZADEH, L.A. , **Fuzzy Sets**, INFORMATION AND CONTROL, 8, 1965, pp. 338–353.

ZADEH, L. A., **A Theory of Approximate Reasoning**, in J.E. Hayes, D. Mitchie and L.I. Mikulich (Eds.) Machine Intelligence, Vol. 9, WILEY, New York, 1979, pp. 149–94.

ZADEH, L. A. , **Soft Computing and Fuzzy Logic**, IEEE SOFTWARE, 1994a, pp. 48–56.

ZADEH, L. A., **Why the Success of Fuzzy Logic Is Not Paradoxical,** IEEE EXPERT - A FUZZY LOGIC SYMPOSIUM, 9(4), 1994b, pp. 43–46.

APPENDIX

Linear Controller is A Subset of A Fuzzy Controller (Proof)

The fuzzy controller function $y = f(x)$, modelled by a fuzzy rule base, can emulate the linear controller:

$$y = \sum_{i=1}^{N_X} c_i \cdot x_i + d = c^T \cdot x + d$$

when meeting the assumptions A1 to A8 introduced in Section 3.

Indeed, according to A1, each fuzzy partition contains only normal and convex fuzzy sets. Consequently, at any value $x_i' \in X_i$ there are no more than two consecutive overlapping fuzzy sets $A_{i,\ell}$ and $A_{i,\ell+1}$, such that (see Figure 1):

$$\mu_{A_{i,\ell}}(x_i') > 0 \quad \text{and} \quad \mu_{A_{i,\ell+1}}(x_i') > 0$$

Furthermore, the number of all contributing fuzzy rules is 2^{N_X}, where N_X is the number of inputs.

Let $\alpha_{i,\ell}$ and $\alpha_{i,\ell+1}$ be the degrees of matching between $A_{i,\ell}$ and $A_{i,\ell+1}$ on the one hand and the crisp input x_i' on the other hand. As a result of A2, we have $\alpha_{i,\ell} + \alpha_{i,\ell+1} = 1$.

Given the aggregation is a summation, the fuzzy controller output $y' = c^T \cdot x' + d$ is described by applying the fuzzy-mean defuzzification directly by using the consequents of all 2^{N_X} contributing fuzzy rules:

$$y' = \frac{\sum_{k=1}^{2^{N_X}} \beta_k b_k}{\sum_{k=1}^{2^{N_X}} \beta_k}$$

where b_k is the crisp consequent (or, in other words, the support of the fuzzy singleton representing the consequent) of fuzzy rule r_k.

Given assumptions A2 to A4 and A6 we can write:

$$\sum_{k=1}^{2^{N_X}} \beta_k = \sum_{k=1}^{2^{N_X}} \prod_{i=1}^{N_X} \alpha_{ik} = \left(\alpha_{1,\ell} + \alpha_{1,\ell+1}\right) \cdot \sum_{k=1}^{2^{N_X-1}} \prod_{i=2}^{N_X} \alpha_{ik} = \ldots$$

$$= \prod_{i=1}^{h} \left[\left(\alpha_{i,\ell} + \alpha_{i,\ell+1}\right) \cdot \sum_{k=1}^{2^{N_X-h+1}} \prod_{i=h+1}^{N_X} \alpha_{ik} \right] = \ldots$$

$$= \prod_{i=1}^{N_X} \left(\alpha_{i,\ell} + \alpha_{i,\ell+1}\right) = \prod_{i=1}^{N_X} 1 = 1$$

Still to prove is:

$$\sum_{k=1}^{2^{N_X}} \beta_k b_k = c^T \cdot x' + d$$

The left-hand side of the above equation can be written as:

$$\sum_{k=1}^{2^{N_X}} \beta_k b_k = \sum_{k=1}^{2^{N_X}} \beta_k \left(c^T \cdot x_k + d\right) =$$

$$= c^T \cdot \sum_{k=1}^{2^{N_X}} \beta_k \cdot x_k + \sum_{k=1}^{2^{N_X}} \beta_k \cdot d =$$

$$= c^T \cdot \sum_{k=1}^{2^{N_X}} \beta_k \cdot x_k + d$$

Hence, the proof is reduced to proving:

$$\sum_{k=1}^{2^{N_X}} \beta_k \cdot x_k = x'$$

For the i^{th} input this can be written as:

$$\sum_{k=1}^{2^{N_X}} \beta_k \cdot x_k = \sum_{k=1}^{2^{N_X}} \prod_{i=1}^{N_X} \alpha_{ik} \cdot x_{ik} =$$

$$\sum_{k=1}^{2^{N_X-1}} \left[\alpha_{i,\ell} \cdot x_{i,\ell} \prod_{\substack{h=1 \\ h \neq i}}^{N_X} \alpha_{hk} + \alpha_{i,\ell+1} \cdot x_{i,\ell+1} \prod_{\substack{h=1 \\ h \neq i}}^{N_X} \alpha_{hk} \right] =$$

$$= \left(\alpha_{i,\ell} \cdot x_{i,\ell} + \alpha_{i,\ell+1} \cdot x_{i,\ell+1}\right) \cdot \sum_{k=1}^{2^{N_X-1}} \prod_{\substack{h=1 \\ h \neq i}}^{N_X} \alpha_{hk} =$$

$$= \left(\alpha_{i,\ell} \cdot x_{i,\ell} + \alpha_{i,\ell+1} \cdot x_{i,\ell+1}\right)$$

which is equal to x_i', because $\alpha_{i,\ell+1} = 1 - \alpha_{i,\ell}$ and:

$$\alpha_{i,\ell} = \frac{x_{i,\ell+1} - x_i'}{x_{i,\ell+1} - x_{i,\ell}}$$

That means that any linear controller can be emulated by a fuzzy controller represented by a static function description.